EPFL –ISR- Groupe VRAI

Diploma Thesis
Academic year 2001-2002

Detection, tracking and registration of shapes with a vision system

Student *Emilio Casanova*

Professor *Prof. Roland Siegwart*

Assistants *Sebastien Grange*

Terry Fong

Dr. Charles Baur

Detection, tracking and registration of shapes with a vision system

Emilio Casanova, Micro-engineering

Assistant 1: *Sebastien Grange*
Assistant 2: *Terry Fong, Charles Baur*
Professor: *Roland Siegwart*

Sensor fusion can be used to recover 3D pose and to analyze spatial configuration of object parts. The goal of this project is to develop a sensor fusion method using vision sensors (stereo cameras, color vision). This project addresses the visual tracking problem for a rigid-articulated object using a known geometrical and kinematical model. This model provides us with useful information about its motion constraints. For fast computation, the model is built with elliptical primitives.

This project describes the implementation of a vision-based object tracker, which uses prior knowledge from an object of interest including color, shape and spatial configuration. The tracker uses multiple modalities derived from stereovision and color images. The object is first defined through an interactive interface that allows the user to build an ellipse-based model of it.

The goal of this project is first to develop and implement a real time algorithm for articulated object detection given its ellipse model. The data is coming from a video stream of a digital stereoscopic camera. The found objects are characterized (principle axes, area) and tracked in real time. Then we combine detection with object model. This model is used to constraint the search of many ellipses around which the model will be adjusted. To do so, we use two levels of tracking: MODEL TRACKING and LOCAL TRACKING. The first level is based on object part recognition and on object lost part recovery. The second level is based on simple binary correlation and window prediction.

An interactive interface has been implemented (fig. 1) and detection and tracking of multiple objects has been performed (fig.2).

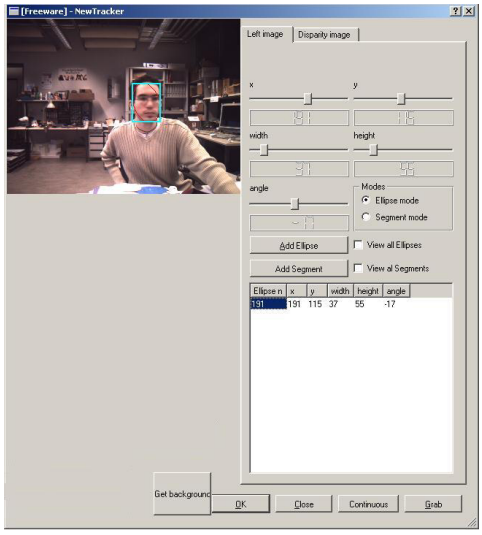


Fig. 1 – User interface

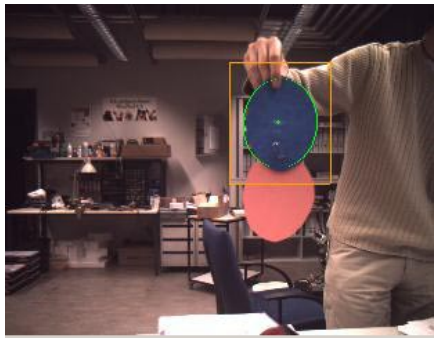


Fig. 2 - Tracking example

Table of contents

List of figures

Abstract

Sensor fusion can be used to recover 3D pose and to analyze spatial configuration of object parts. The goal of this project is to develop a sensor fusion method using vision sensors (stereo cameras, color vision). This project addresses the visual tracking problem for a rigid-articulated object using a known geometrical and kinematical model. This model provides us with useful information about its motion constraints. For fast computation, the model is built with elliptical primitives.

This project describes the implementation of a vision-based object tracker, which uses prior knowledge from an object of interest including color, shape and spatial configuration. The tracker uses multiple modalities derived from stereovision and color images. The object is first defined through an interactive interface that allows the user to build an ellipse-based model of it.

1 Introduction

Shape detection is one of the fundamental tasks in computer vision systems. It can be used to reinforce color-based object detection for identification and registration. The use of primitive models (shape) allows a reduction and a simplification of data and, consequently, leads to faster and simpler processing. A very important primitive is an ellipse, which, being a perspective projection of a circle, is used in many applications of computer vision like 3D vision and object recognition, medical imaging, etc

The goal of this project is first to develop and implement a real time algorithm for articulated object detection given its ellipse model. The data is coming from a video stream of a digital stereoscopic camera. The objects found are characterized (principle axes, area) and tracked in real time. Then we combine detection with object model. This model is used to constraint the search of many ellipses around which the model will be adjusted. To do so, we use two levels of tracking: MODEL TRACKING and LOCAL TRACKING. The first level is based on object part recognition and on object lost part recovery. The second level is based on simple binary correlation and window prediction. A possible application is the detection and tracking of the complete human body.

This report is divided into 7 sections. Section 2 gives the context of this project and the modalities. Current research that is relevant to this project is described in section 3. Section 4 describes our tracking design. The implementation is made in section 5. The following section 6 presents the results. Finally, the future possible development is discussed in section 7.

2 Context

2.1 Project specifications

This project aims at using vision-based sensor fusion to create a tracker that can detect, track and register shapes. It consists of three main parts:

- object modeling
  design an interactive interface in order to decompose the object of interest in several ellipses and generate an ellipse-based model out of it with color, size and spatial characterization. The model is hierarchical.

- object detection
  combine image processing techniques such as stereo vision and color processing to achieve robust detection (identification).

- object tracking and registration
  combine the parameters provided by the detection phase with the information generated by the modeling phase to achieve robust tracking.

2.2 Design hypothesis

Given the specifications described above, there are several technical design constraints that will affect the design of our tracking system:

- real-time
  obviously, since our tracker deals with 3D pose recovery, it has to be a real-time application; the software will be optimized to minimize the time loop of the vision process

- single object tracking
  because we provide only one model of the object of interest and the performance will decrease if a large amount of object are

- stationary camera, moving objects (SCMO)
  - background subtraction in tracking for more reliable tracking

- flexible
  should be flexible enough to be easily enhanced with more feature at a later time

2.3 Theoretical overview

To reach our goal we have to use several image processing tools such as stereo vision, color filtering and ellipse fitting.

2.3.1 Stereo imaging

Calculating the distance of various points in the scene relative to the position of the camera is one of the difficult tasks for a computer vision system. A common method for extracting such depth information from intensity images is to acquire a pair of images using two cameras displaced from each other by a known distance (baseline). The geometry of binocular stereo is shown in figure 2.1.

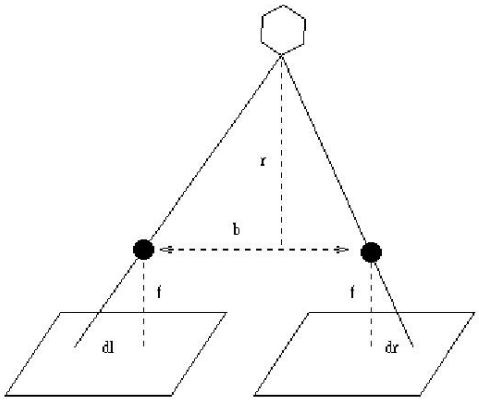


figure 2.1 - Calculating depth from disparity in stereovision

b: baseline

d: disparity

f: focal length

r: depth

$$with\ r = \frac{b \cdot f}{d} \quad (1)$$

where d = dl-dr

The technique used in this project is based on area correlation method. A diagram of the stereo system is given in figure 2.2.

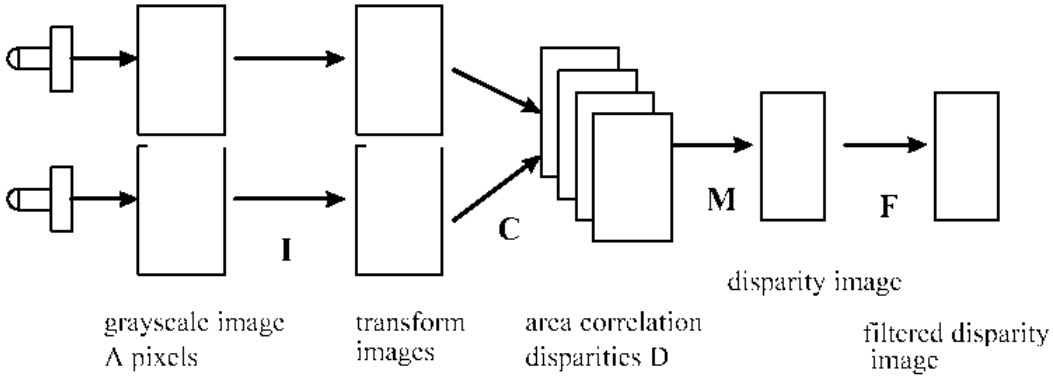


figure 2.2 - Stereovision system

In the first step two images are grabbed from the cameras (I); then the images are transformed for correcting the optical aberrations (C). In the next step is made the area correlation, which gives the disparity map (M), and finally this map is filtered for errors (F) [10].

Benefits

There are many features of stereo data that can be used to track objects. The first is that stereo images allow retrieving shapes independently from the color. With this technique we can segment objects from the scene but also analyze the shape of multicolor object.

The second feature is that we can calculate the real-world size of the objects. If the distance to an object in known, one can extrapolate its real size from it size in the image. This makes object recognition more robust. Real size is a strong hint for discriminating when trying to match a model.

Finally, stereo can be very useful in the tracking of occluded objects. Range information allows us to see if an object is in front of or behind another.

figure 2.3 - Left color image

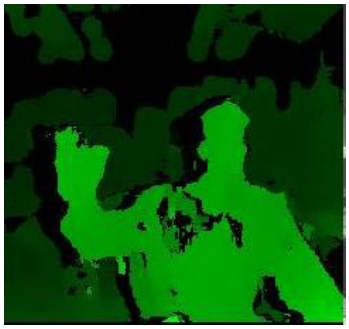


figure 2.4 - Computed stereo

Limitations

Several problems can emerge such as the sensibility to calibration and the lack of texture in the objects. Calibration is very important because it gives the internal and external parameters allowing to extract accurate depth information. The lack of texture implies that the algorithm can't find any correspondence of a given point in the pair of images. We have to say that disparity processing can be a noisy process. Meaningful data is only available in the horopter region (region between the two planes in figure 2.5). It implies that the object of interest has to be in a given region in front of the camera. The fact that the relationship between range and disparity is non linear (figure 2.6) implies that a given object will not spread over the same number of disparities if it's closer or further of the camera.

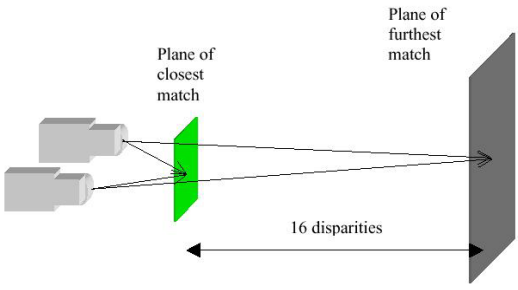


figure 2.5 - Horopter region

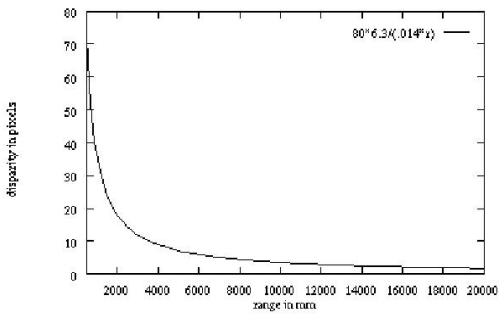


figure 2.6 - Relation between range [mm] and disparity [pixel]

2.3.2 Modeling

There are several modeling solutions for describing shapes and their spatial organization:

3D models

In figure 2.7 we can see 3D models arranged hierarchically, each one based on a spatial configuration of a few sticks or axes, to which volumetric or surface shape primitives are attached.

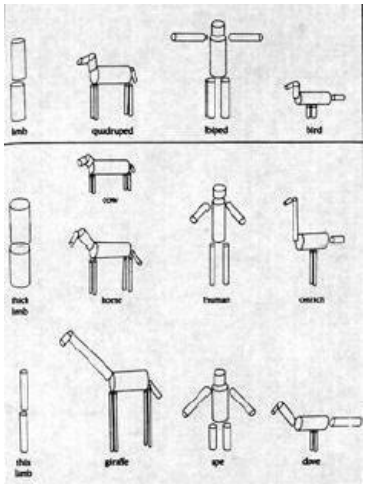


figure 2.7 - Modular hierarchical representation including volumetric and surfaces primitives

Generalized cylinders

Marr's 3D model shows that we can decompose a 3D shape description into lower order problems. It also provides a framework for hierarchical representation (figure 2.8).

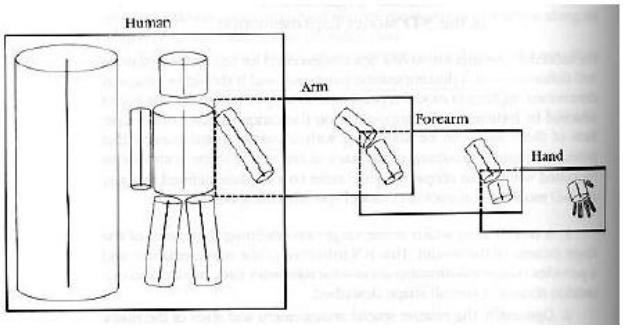


figure 2.8 - Marr's 3D model

Primitives

Quadrics are useful primitives for describing shapes such as cones, cylinders, planes, ellipsoids, hyperboloids, and paraboloids. They are surfaces of the algebraic type:

$\sum\sum\sum_{i+j+k\le 2} a_{ijk} x^i y^j z^k = 0$ Where a_{ijk} are the coefficients.

An example in figure 2.9 is the superellipse (n>2) or subellipse (n<2) given by the equation below :

$$\frac{|x|^n}{a^n} + \frac{|y|^n}{b^n} = 1$$

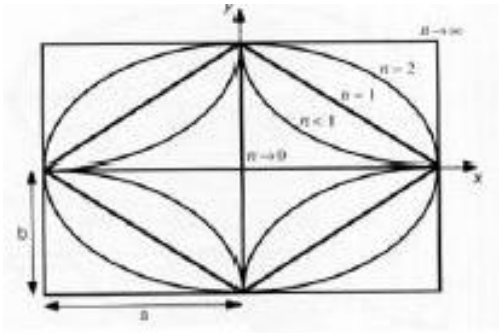


figure 2.9 - Illustration of the superellipses and the subellipses

Ellipse is particularly interesting because this form is invariant by rotation, translation and scaling and is therefore a good primitive for object modeling.

2.3.3 Ellipse fitting

In this project, as it was explained before, the ellipse is a good primitive to use for object modeling. Thus we have explored several methods for ellipse detection and fitting. There are two main approaches to that problem: voting/clustering and optimization methods. In the first group we have Hough transform and fuzzy clustering. They are robust against outliers and can detect multiple primitives at once, but these methods are unfortunately slow and require a large amount of memory. The second group of fitting methods, among which we can find the least square approach are based on optimization of an objective function which characterizes a goodness of a particular ellipse with respect to the given set of data point. The main advantages of these methods are they speed and accuracy, but they can fit only one primitive at time. Also the sensibility to outliers is higher than in the clustering methods [7].

For this project we use a method of the second category called the "Direct ellipse-specific fitting" method. The standard equation of a conic is:

$$F(x, y) = ax^2 + bxy + cy^2 + dx + ey + f = 0 \quad (2)$$

with an ellipse-specific constraint

$$b^2 - 4ac < 0 \quad (3)$$

where a, b, c, d, e, f are the ellipse
and (x, y) are coordinates of points lying on it.
By introducing vectors

$$\vec{a} = [a, b, c, d, e, f]^T \quad \vec{x} = [x^2, xy, y^2, x, y, 1] \quad (4)$$

it can be rewritten to the vector from

$$F_{\vec{a}}(\vec{x}) = \vec{x} \cdot \vec{a} = 0 \quad (5)$$

figure 2.10 - Fitting of six points with the "Direct ellipse-specific fitting" method

Given a set of points (x_i, y_i), $i = 1 \ldots N$ the algorithm finds $\vec{a}$ that respect the ellipse-specific constraint given by (3)

2.3.4 Color Spaces

The term “Color Space” refers to the way colors are represented. Color is very useful to filter the image for keeping only relevant data. But we have to keep in mind the computational effort for changing of color space. In table 1 we give the common used color spaces.

Color Space	dimension	Components
Gray	1	intensity
normalized color rgb	2	r normalized red g normalized green
RGB	3	R red intensity G green intensity B blue intensity
YUV	3	Y luminance U chrominance V chrominance
HIS	3	H hue S saturation I intensity

table 1 - commonly used color spaces

RGB Color Space

The RGB color space is perhaps the most commonly used color description method. It can be represented as a cube, with the three axes representing Red, Green and Blue components respectively. The intensity varies from zero (black) to the maximum value (white) along the diagonal of the cube (figure 2.11).

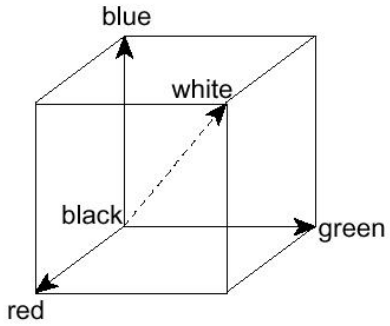
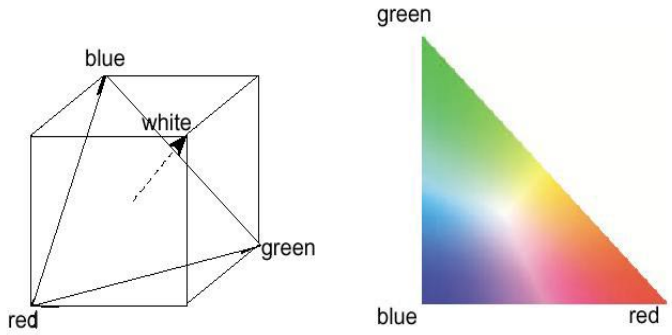


figure 2.11 - The RGB Cube with intensity diagonal

rgb Color Space

The rgb or Chromaticity color space is an intensity invariant color space. To convert from RGB to rgb a simple normalization is performed by dividing each component by the sum. This space is very interesting because the computational effort is very low (only 1 division and 2 additions per channel).



$$r(R,G,B) = \frac{R}{R+G+B} \quad (6)$$

$$g(R,G,B) = \frac{G}{R+G+B} \quad (7)$$

$$b(R,G,B) = \frac{B}{R+G+B} \quad (8)$$

figure 2.12 - (Left) The rg-Calor Space, found by projecting RGB-value to a plane perpendicular to the intensity axis. (Right) Representation of rgb in Chromaticity coordinates.

HIS Color Space

The HIS Color Space represents the visible colors by their hue, saturation and intensity. This model takes advantage of the way humans perceive color. Hue (H) is the color perceived due to the wavelength. Saturation (S) is the degree to which the color is pure and not polluted with white light (grayscale). High saturation means that the color is highly pure, while low saturation means that the color is highly diluted with white light (RGB of equal amounts, or grayscale). H and S together provide the *chromaticity*, that is, color content of a pixel. Intensity is brightness.
Thus we have:

- hue color, from red to blue
- saturation the purity of the color
- intensity intensity, or brightness due to energy level

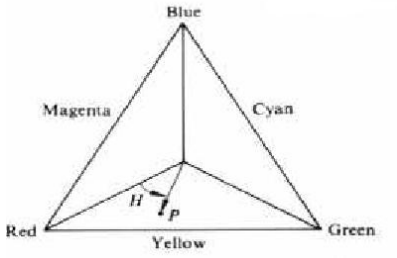


figure 2.13 - Representation of the HIS Color Space

However the HIS also has some inconveniences: it requires more computational effort that the rgb Color Space. When converting from RGB space and for very low intensities the hue becomes unstable. This is due to the quantization of the RGB cube. Also Hue and Saturation are not defined at $R = G = B = 0$.

3 Related research

3.1 Tracking

There are a lot of projects dealing with multi-object and articulated objects tracking. They addressed similar problems involving tracking of a known model. Here is a selection of some relevant ones:

3.1.1 Object posture modeling

A part of this project consists in performing some level of object modeling. The modeling should be accurate enough to allow meaningful posture description, while simple enough to be performed in real-time.

There are several techniques to track multiple objects such as model-based methods and multi-layer analysis. Yunqiang Chen and Thomas S. Huang [1] include in their work these two trends into a Map framework for tracking objects (human hands or faces).

This method is based over a hierarchical Markov Random Field (MRF) model (figure 3.1). They assume that each object (hands, faces) can be approximated by a coherent color blob and use a mixture of multivariate Gaussians to encode the color value, the centroid and second moments. They work in the YUV color space and use conditional probabilities (likelihood function) to classify every pixel in current frame to a blob. The main difference with this present work is that they don't use range information and thus don't have real object size. The fact using YUV color space makes tracking non-intensity invariant. Thus it could be some problems with variable lightning conditions

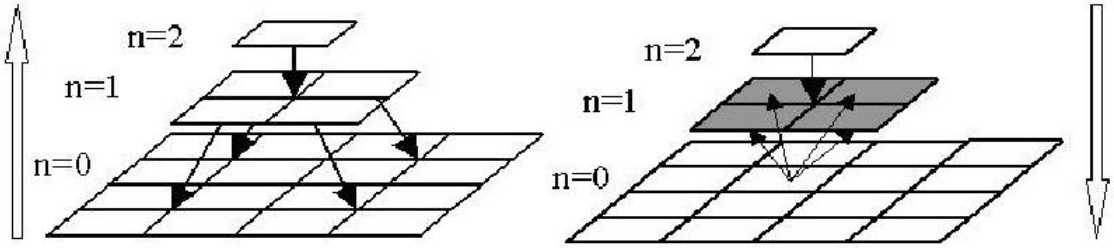


figure 3.1 - Structure of Hierarchical MRF Model

figure 3.2 - Results of head and hand tracking

Michael H. Lin [2] propose in his work to model each target segment as a planar patch bounded by the convex hull of two circles (figure 3.3), and both edge-like and region-like information in matching the model to the target. The main difference between this paper and this project is that they don't use intensity or color information and need initial conditions.

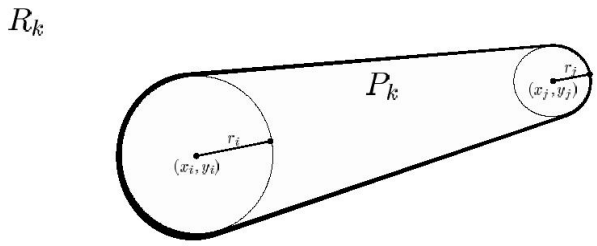


figure 3.3 - Segment of the target modeled by a planar patch

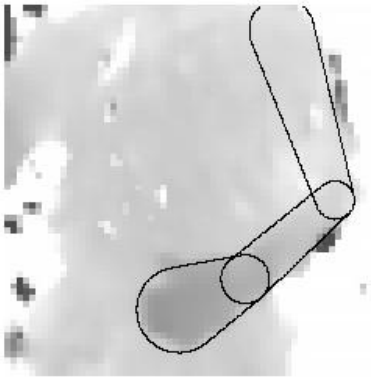


figure 3.4 - Estimated target pose overlaid on the input range image sequence. Outlined regions represent the estimated positions of the hand, forearm, and upper arm.

Another interesting work from N. Jojic, M. Turk and T. S. Huang [9] use a statistical image formation model made of sticks.

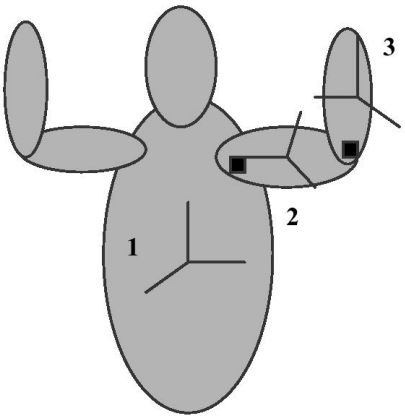


figure 3.5 – articulated model – the local coordinate systems and joints

figure 3.6 – stick-based human body modeling

Liang Zhao and Chuck Thorpe [3] propose a recursive context reasoning approach (RCR in figure 3.7) to detect human and identify body parts using a TRS (translation, rotation, scaling) invariant probabilistic model to encode the shapes of the body parts including the size and spatial relationships between them (figure 3.8). They use a Bayesian framework to perform human detection and part identification under partial occlusion. But there are some limitations due to the inherent ambiguity with contour features. We can see some dissimilarity with the present project concerning the modeling step because our modeling step is object specific. On the other hand they use a deformable model with joint Gaussian distribution.

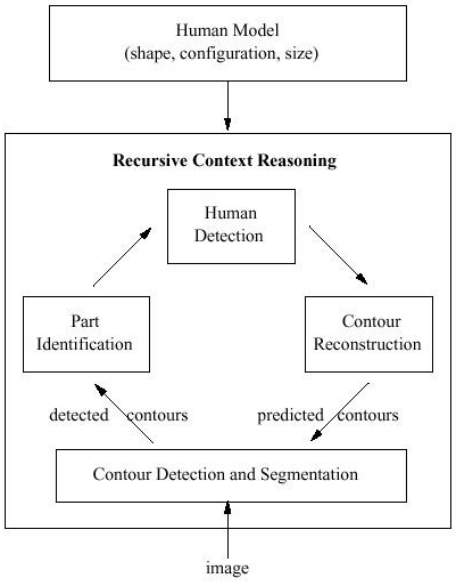


figure 3.7 - Flow chart of the RCR algorithm

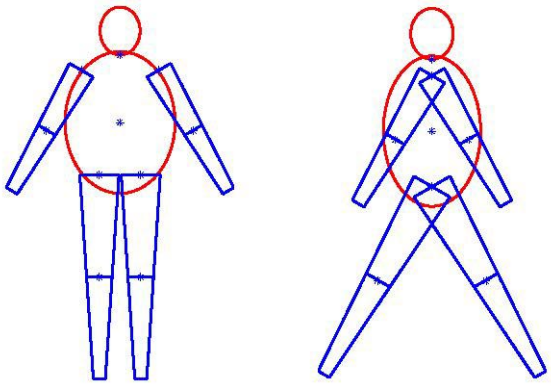


figure 3.8 - Human body model.(Left) front view and (Right) side view

Another interesting work is given by Lars Bretzner [4] in which he explain the power of a multi-scale feature hierarchies for object tracking. And use different cues such as patch similarity. This measure is a normalized Gaussian-weighted intensity cross-correlation between two image patches. They consider several types of qualitative relations such as spatial coincidence, directional relation. The differences with the present work are that they don't use color information and don't work with real object size. In another way they work in specific background conditions.

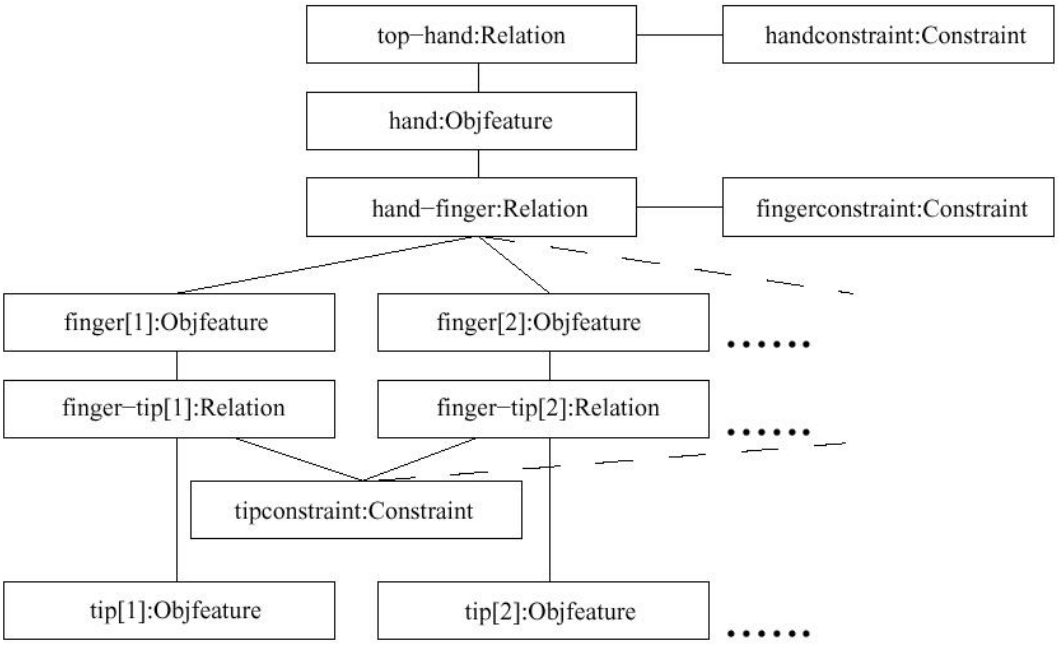


figure 3.9 - Instance diagram of the feature hierarchy of a hand show in figure 3.10

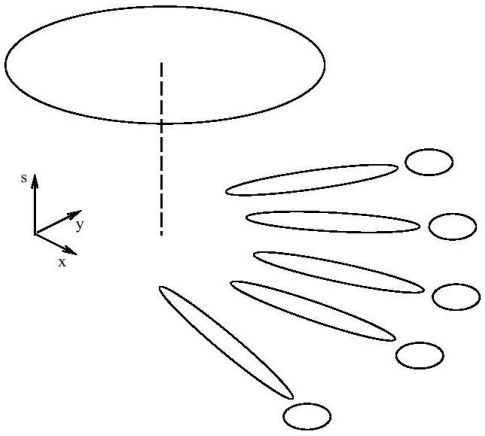


figure 3.10 - A qualitative multi-scale feature hierarchy constructed for a hand model

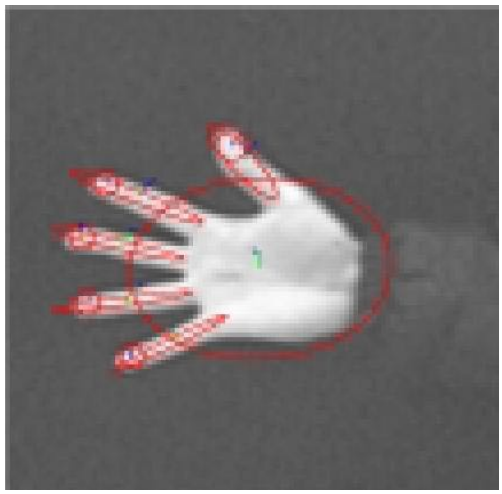


figure 3.11 - All hand features captured

4 Tracker design

4.1 Overview

Our tracking architecture is based on the successive integration over 3 layers, as illustrated in figure 4.1.

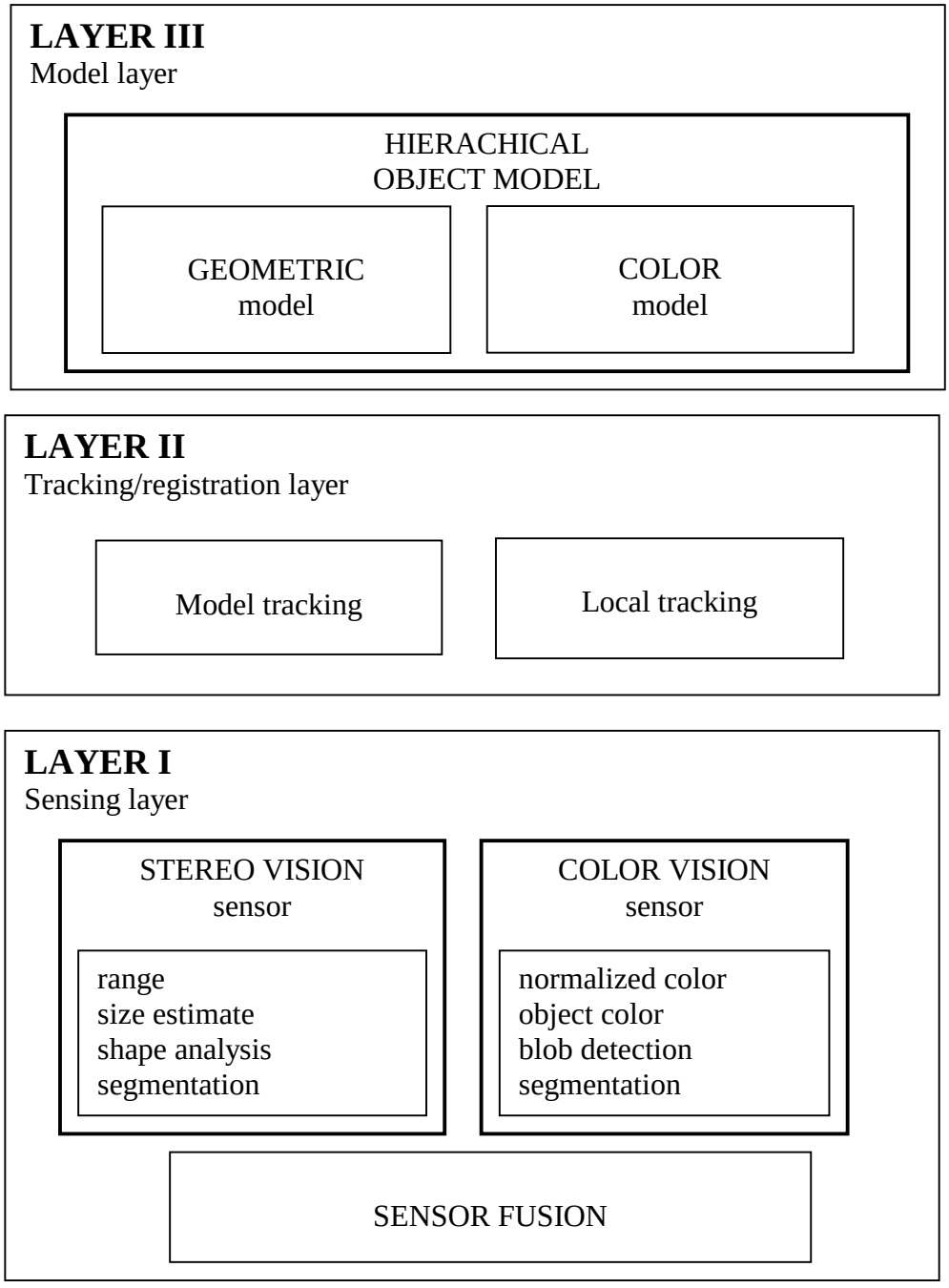


figure 4.1 - Integration layers

Layer I: sensing

The first layer integrates the color and stereo processing. It locates meaningful data for post-processing in the tracking and registration layer.

Layer II: tracking and registration

The second layer integrates the information provided by the sensing and modeling layer using a tracking strategy with two modes:

- Model tracking (Locate lost object part with the model information)
- Local tracking (Locate and track an object part knowing its position in the previous frame)

Layer III: modeling

The final layer could be described as the modeling layer. It combines all of the available inputs and processes so that the user can construct an ellipse-based model with color and stereo modalities. Color is a useful hint for object recognition. Stereo allows computing the real size of the objects.

4.2 Object modeling

An interactive interface figure 4.2 has been developed allowing the user to grab an image and to select the parts of the object he wants to track. One ellipse models each part. All the ellipses are connected with line segments to create the full model.

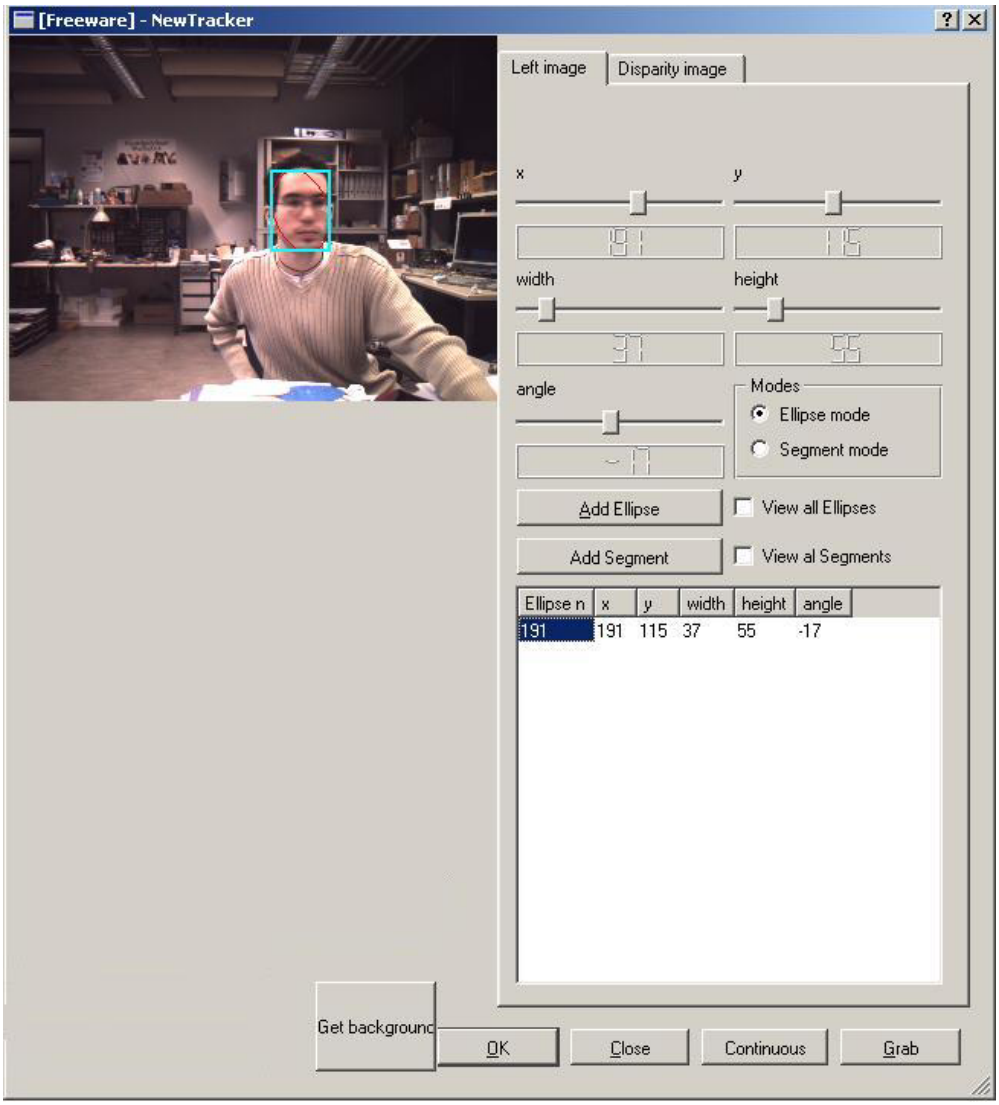


figure 4.2 - Modeling interface

The model includes for each ellipse:

- color information compute from R, G, B channels
- shape information (size, ratio)
- depth information compute from area correlation algorithm
- position (in the image frame)
- hierarchical structure with parent ellipse and child ellipse connected with line segments (each segment as is motion range in the image plane)

For each object we can compute a color histogram. We then calculate:

$$Nr_{\max} = \max\left(\frac{R}{R+G+B}\right) \ , \ SigmaLeft(Nr_{\max}) , SigmaRight(Nr_{\max})$$

$$Ng_{\max} = \max\left(\frac{G}{R+G+B}\right) \ , \ SigmaLeft(Ng_{\max}) , SigmaRight(Ng_{\max})$$

SigmaLeft (SL) and SigmaRight (SR) are respectively the left and right distance of the N_{max} pic.

With the Sigma values we can compute a color filter (figure 4.3). This provides a color signature for the object.

Filter=[minColor,maxColor]

The minimum colors of the filter are:

- Nr_{max}- SigmaLeft(Nr_{max}) and Ng_{max} - SigmaLeft(Ng_{max}).

The maximum colors are:

- Nr_{max}+ SigmaRight(Nr_{max}) and Ng_{max} + SigmaRight(Ng_{max})

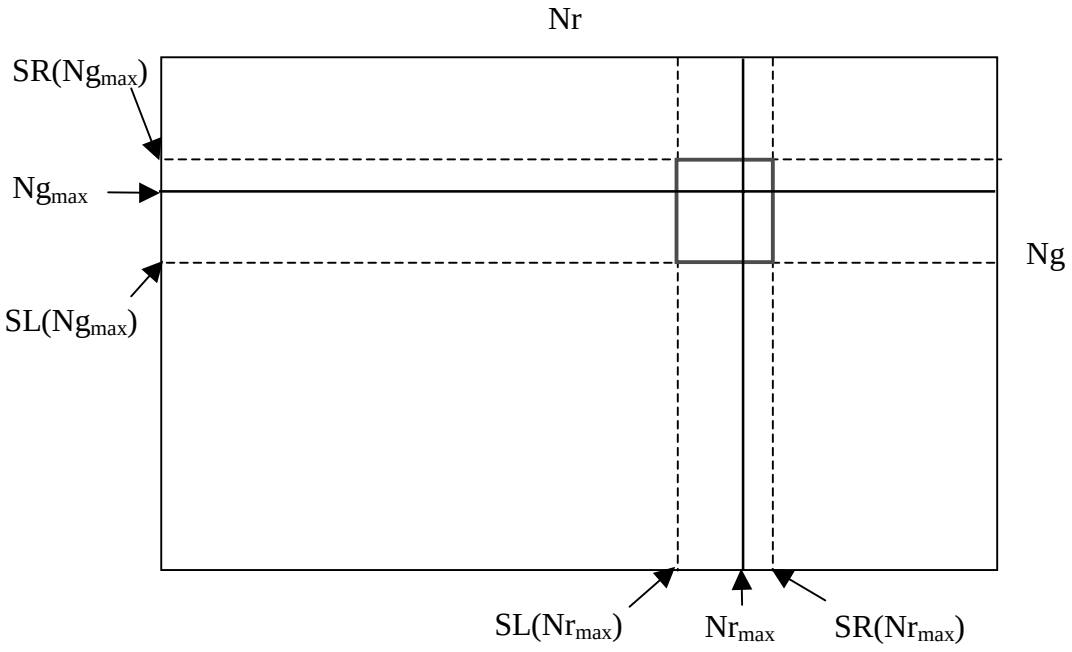


figure 4.3 - Filter in the normalized color space

Here is an example of object modeling. In figure 4.4 the user as selected the head then she is modeled by an ellipse (figure 4.5). In the next step the histogram in normalized colors is compute in the ellipse area (figure 4.7). Finally a color filter is designed from the histogram and the image of figure 4.4 is filtered (figure 4.6).

figure 4.4 - Left image with head ellipse (red) and bounding box (blue)

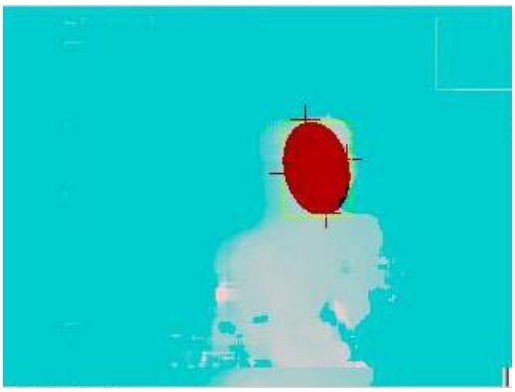


figure 4.5 - Stereo image of the foreground with head ellipse(red)

figure 4.6 - Filtered image

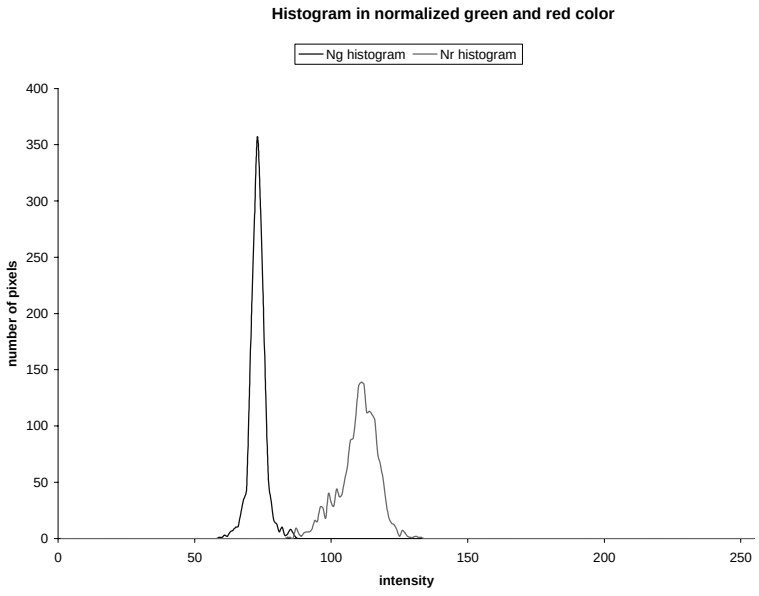


figure 4.7 - Histogram in normalized green and red color

4.3 Object detection

The goal of the detection step is to decrease the quantity of data to process. First, according to the object / ellipse properties (given by the model) we process only a given area (search area) limited by the motion constraint of the child object (figure 4.8). Then we apply successively a color filter and a range filter.
The range filter is compute on the foreground stereo image, which is calculated with the background image and the stereo image. We have implemented an algorithm that interpolates the non calculated values of the disparity image. Therefore we have a dense disparity map without holes. But we have to improve the quality and speed of the process.

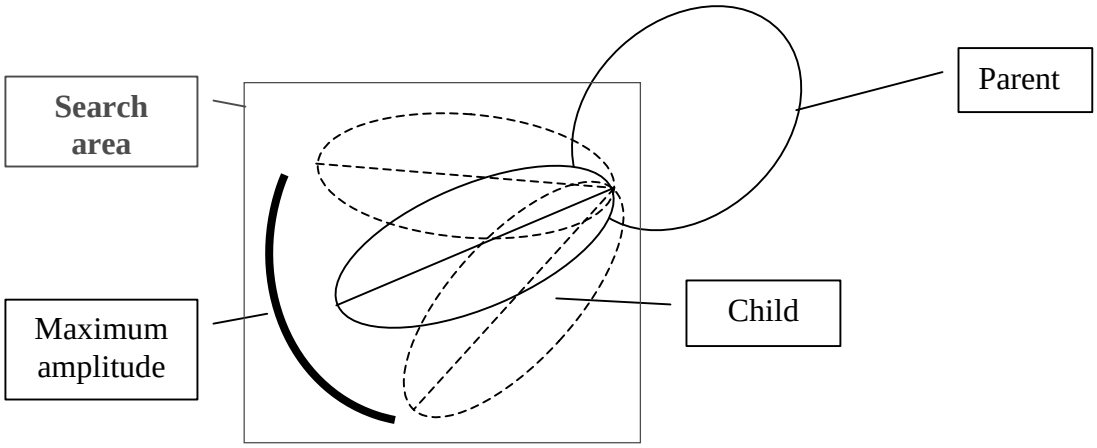


figure 4.8 - Child-Parent motion constraint

At this step we have a binary image with blobs. We make some morphological operation to remove the noise. Then we process the blobs. First we label the blobs, and then we filter the blobs in projected area (minimum and maximum).

After this process, a list of possible objects is available. Then for each object in the list we calculate a best fit ellipse which we compare with the model. The match is made using the real-world size area. For the detection phase, we chose a least square approach for ellipse fitting which is the best suited (see section 2.2.3).

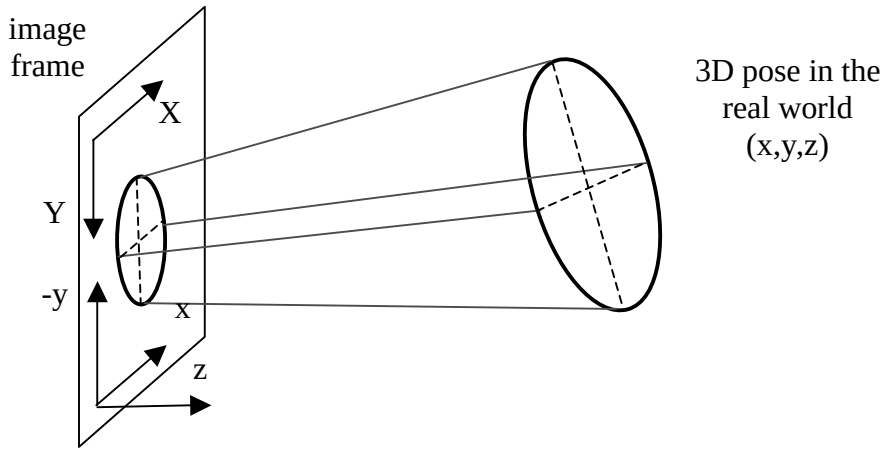


figure 4.9 - 3D pose recovery

We must now describe how we calculate the real world 3D coordinates of the ellipse $[x, y, z]$ given the baseline b, the focal f and the disparity D(X,Y). This is done with the equations below :

$$x = Xz / f \ , y = Xz / f \ , z = bf / D(X,Y) \quad (9)$$

If we have in the image frame an ellipse characterized by:

- (X_0,Y_0) the position of its center
- W its width
- H its height
- PHI its orientation

We have to compute the points (x_i, y_i) for i = 1…4 knowing (X_i, Y_i) .

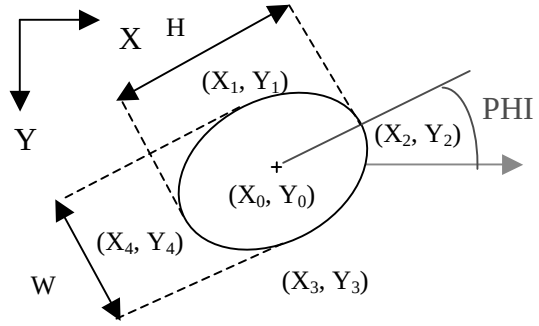


figure 4.10 – Ellipse in the image frame

We have to pay attention with the disparities $D(X_i, Y_i)$ because it's possible that value of the interpolation hasn't consistency with the object. Therefore it is important to verify that their belong to the object. We have to calculate the disparity along the main axes of the ellipse and check the consistency of these values with each disparity $D(X_i, Y_i)$.

4.4 Object tracking and registration

In this section we explain the two modes of tracking:

This approach uses the hierarchical information of the model. The model contains an ellipse structure of parents/children connected with line segments that lets us know the range of position of each ellipse respectively to one another. To search for new ellipses we have implemented two approaches:

- begin search starting with the parent ellipse
- begin search starting with the child ellipse

This combination makes the tracking more robust

4.4.1 Model tracking

The model tracking algorithm looks for an object with a certain color in a given volume of space depending on the geometrical and kinematical constrains given by its parent object (*).

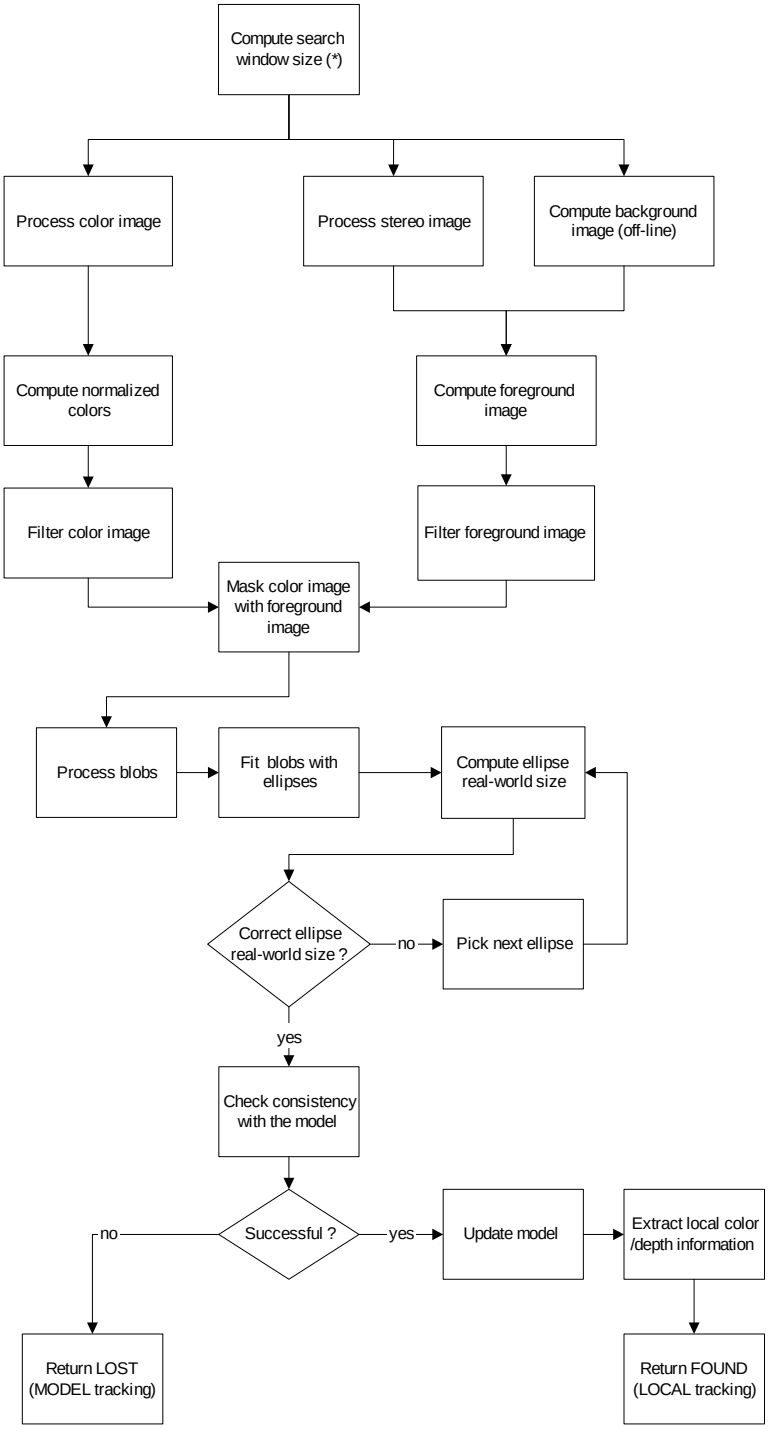


figure 4.11 –Model object tracking

4.4.2 Local tracking

Once the object is located, the images are only processed in an area surrounding the expect position of the object in order to limit processing time. The schematic in figure 4.12 shows the algorithm used in local tracking step.

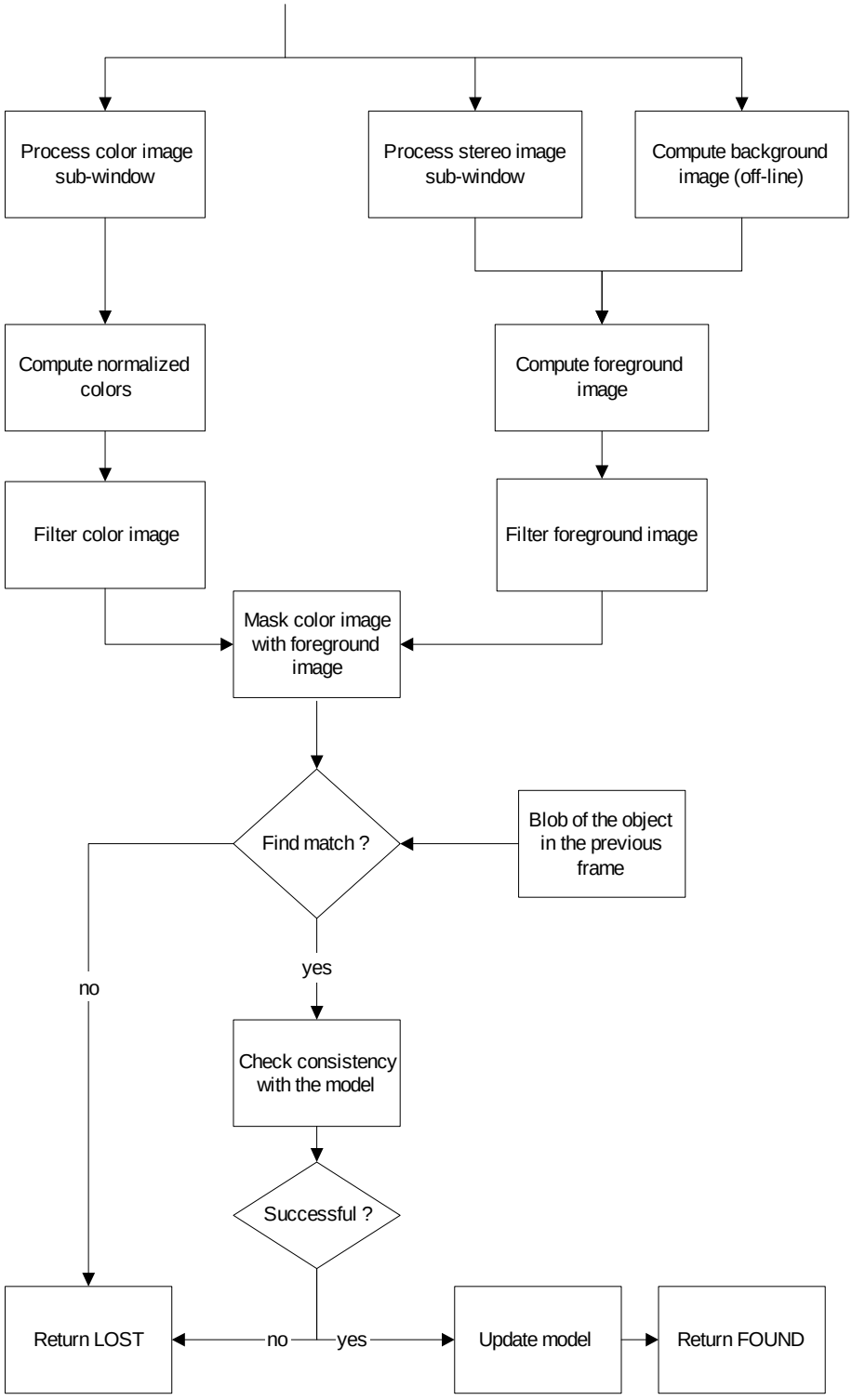


figure 4.12 - Local object tracking algorithm

5 Implementation

This section explain how the tracker is implemented and describes the technical aspects

5.1 Hardware

The hardware platform used to develop our tracker engine is:

- 1 PC Pentium III 700 MHz, 512 RAM
- 2 1.3 Megapixel, progressive scan CMOS imagers
- 1 digital 1394 port

For more technical information go to the appendix

5.2 Structure

The structure of the tracker application in given by the figure

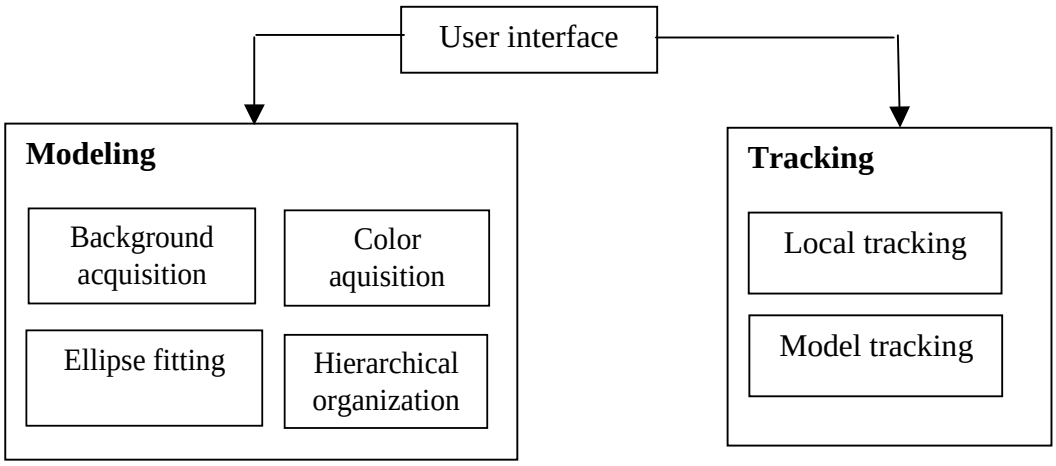


figure 5.1 – Application structure

5.3 Operation

In this part we explain the basics step used in our tracker algorithm:

The object to model is formed by one blue paper ellipse and another pink. The ellipses are attached.

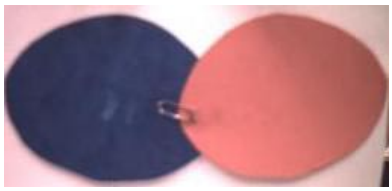


figure 5.2 - Object to model

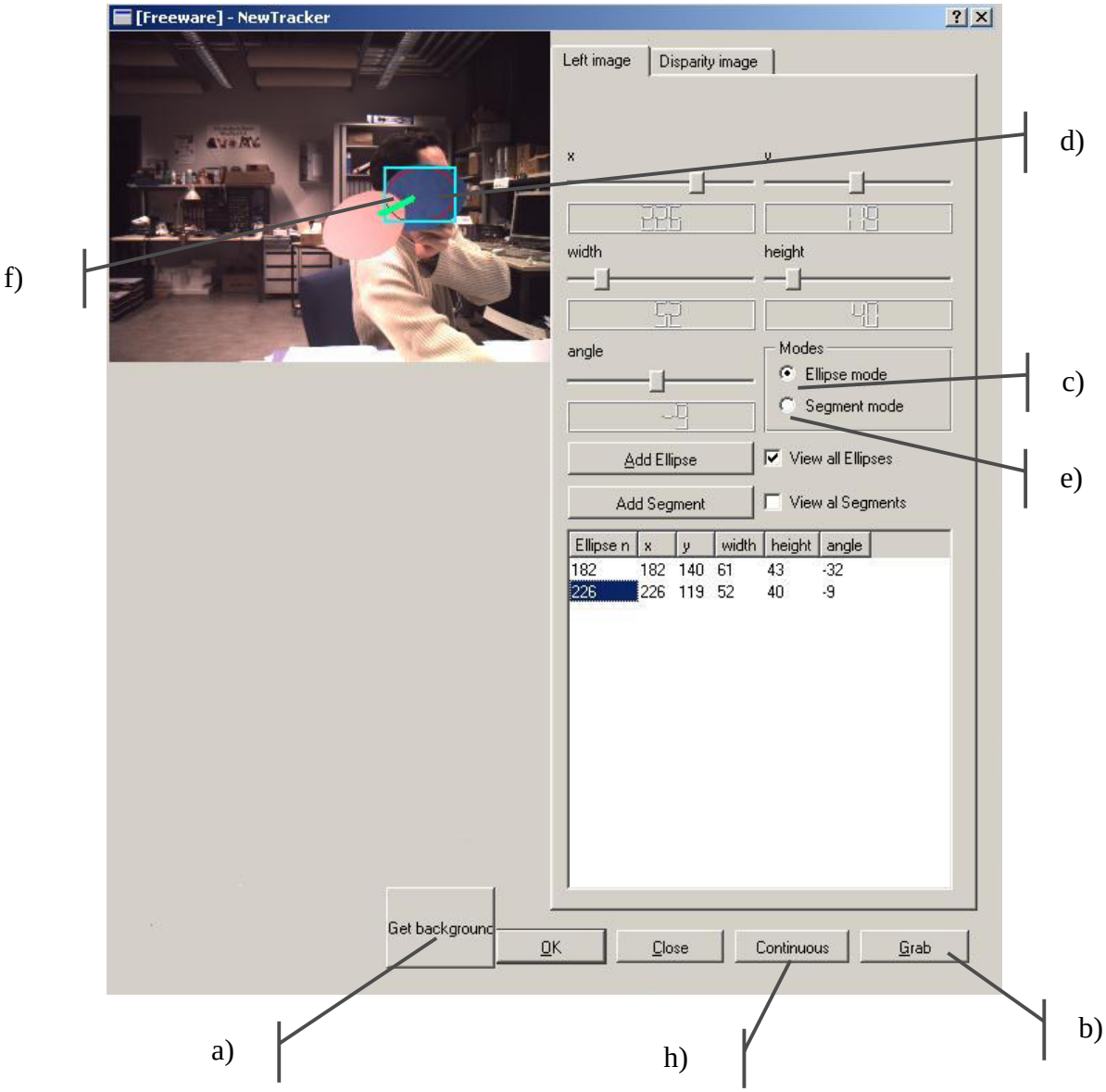


figure 5.3 – User interface

a) Press the “get background” button for making the subtraction background and compute the foreground.

b) Press the “Grab” button to acquire an image of the camera.

c) Press the “Ellipse Mode” radio button

d) Select with the mouse the parts of the object you want to be in the model. The selected part is fit with an ellipse. The ellipse list it automatically updated, the color object is memorized and the real area is compute.

e) Press the “Segment Mode” radio button

f) When all the parts are selected you have to create the model hierarchies by connecting each ellipse respectively to one another. For this operation you have to click with the mouse in the parent ellipse and then in the child ellipse. A line segment appears connecting these two ellipses.

g) For each line segment we can give the maximum amplitude (NOT YET IMPLEMENTED)

h) When the model is ready press the “Continuous” button for activate the images acquisition and the tracking algorithm.

6 Results

6.1 Modeling step

1. Object model acquisition (manually)

2. Foreground for real area computation (automatic)

3. Initialization for the color filter (automatic)

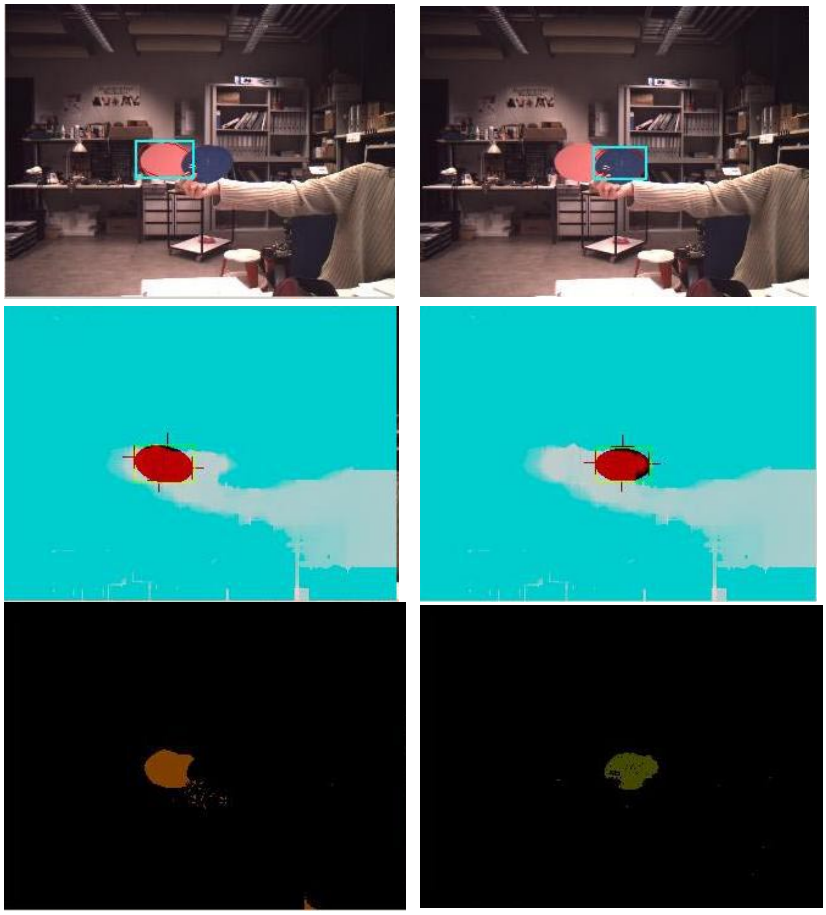


figure 6.1 - Modeling initialization (off-line)

6.2 Model tracking and local tracking

Tracking results for each object part of the model. Outline of the found ellipse in green
The orange box is the search area

figure 6.2 - Model tracking and local tracking

7 Future work

For the moment, the tracker tracks only one part of the model at a time. Thus, the next step will consist in making use of the whole model.

7.1 Limitations

One limitation concerns the color filtering that could be better integrated with the disparity filtering. This color filtering is for the moment not robust enough to variations of lightning conditions.

Another concern lies within the method of interpolation used to remove the holes in the disparity map. This stage slows down the processing loop.

7.2 Improvements

Concerning the color filtering, we could investigate other techniques such as “color indexing” to get an object color signature. We can also improve the procedure of disparity interpolation or use another criterion for disparity processing. One interesting solution is to use the method proposed by H. Tao and H. S. Sawhney *[11*]. This method is based on global matching criterion and color based segmentation that provides reliable depth for thin structures and textureless regions, as well as hypothesizing the correct depth for unmatched regions.

There also exists on the market a new chip that integrates range processing, foreground/background separation, as well as identification, tracking, and occlusion. This chip is produced by company TYZX.

8 Conclusion

This project describes the implementation of a vision-based object tracker, which uses prior knowledge from an object of interest including color, shape and spatial configuration. The tracker uses multiple modalities derived from stereovision and color images. The object is first defined through an interactive interface that allows the user to build an ellipse-based model of it.

The goal of this project was first to develop and implement a real time algorithm for articulated object detection given its ellipse model. The data is coming from a video stream of a digital stereoscopic camera. The found objects are characterized (principle axes, area) and tracked in real time. Then we planned an architecture that would combine detection with object model. This model would be used to constraint the search of many ellipses around which the model would be adjusted. To do so, we are using two levels of tracking: MODEL TRACKING and LOCAL TRACKING. The first level is based on object part recognition and on object lost part recovery. The second level is based on simple binary correlation and window prediction.

For the moment, the tracker tracks only one part of the model at a time. Thus, the next step will consist in making use of the whole model. The current solution includes the user interface, which makes model definition possible. Single object tracking is reliable, and has been implemented towards a future integration that will take model constraints into account.

9 Acknowledgments

I would like to thank Sebastien Grange, and all the members of VRAI- GROUP for their support.

Finally I would like to thank my family and friends, here and abroad, for their permanent, unconditional support throughout my studies.

Lausanne, 22nd of February 2000

Emilio Casanova

10 References

10.1 Research papers

[1] Y. Chen, T. S. Huang, “Hierarchical MRF model for model-based multi-object tracking”, to appear in Proc. IEEE Int’l Conf. on Image Processing 2001, Thessaloniki, Greece.

[2] M. H. Lin, “Tracking Articulated Objects in real-time range image sequences”, International Conference on Computer Vision, September 1999, pages 648-653.

[3] L. Zhao, C. Thorpe, “Recursive Context Reasoning for Human Detection and Parts Identification”, IEEE Workshop on Human Modeling, Analysis, and Synthesis, June 2000.

[4] L. Bretzner, “Multi-Scale Feature Tracking and Motion Estimation”, Dissertation, Computational Vision and Active Perception Laboratory (CVAP), October 1999.

[5] A. W. Fitzgibbon, “Stable Segmentation of 2D Curves”, Ph. D., University of Edingburgh, 1997.

[6] S. Grange, “Vision-based Sensor Fusion for Active Interfaces: H.O.T. – Human Oriented Tracking”, EPFL, 2000.

[7] R. Halíř, J. Flusser, “Numerically stable direct least squares fitting of ellipses”

[8] A. W. Fitzgibbon, M. Pilu, R. B. Fisher, “Direct Least Squares Fitting of Ellipses”, Departement of Artificial Intelligence, January 4, 1996.

[9] N. Jojic, M. Turk, T. S. Huang, “Tracking Self-Occluding Articulated Objects in Dense Disparity Maps”, IEEE International Conference on Computer Vision, Corfu, Greece, September 1999.

[10] K. Konolidge, “Technical Notes: Stereo Geometry”, SRI International, August, 1999.

[11] H. Tao, H. S. Sawhney, “Global Matching Criterion and Color Segmentation Based Stereo”, in Proc. Workshop on the Application of Computer Vision (WACV2000), pp. 246-253, December 2000.

10.2 Lectures

“Robust vision for vision-based control of motion”, edited by M. Vincze, G. D. Hager, SPIE/IEEE series on imaging science & engineering, 2000.

A. Jaklič, A. Leonardis, F. Solina, “Segmentation and recovery of superquadrics”, Computational imaging and vision, Kluwer Academic Publishers, 2000.

D. Scharstein, “View synthesis using stereo vision”, Lecture notes in computer science; vol. 1583, Springer, 1999.